A Layer 2.5 Toolkit

Mark Carson

NIST

April, 2002

http://www.antd.nist.gov/nistswitch/linux/

Outline

- Why a layer 2.5 toolkit? *(Past)*
- Why a new NIST Switch? *(Present)*
  - Architecture
  - Demonstration of "interesting" features
- What can NIST Switch enable? *(Future)*
  - Optimized multipath in "fairly flat" networks

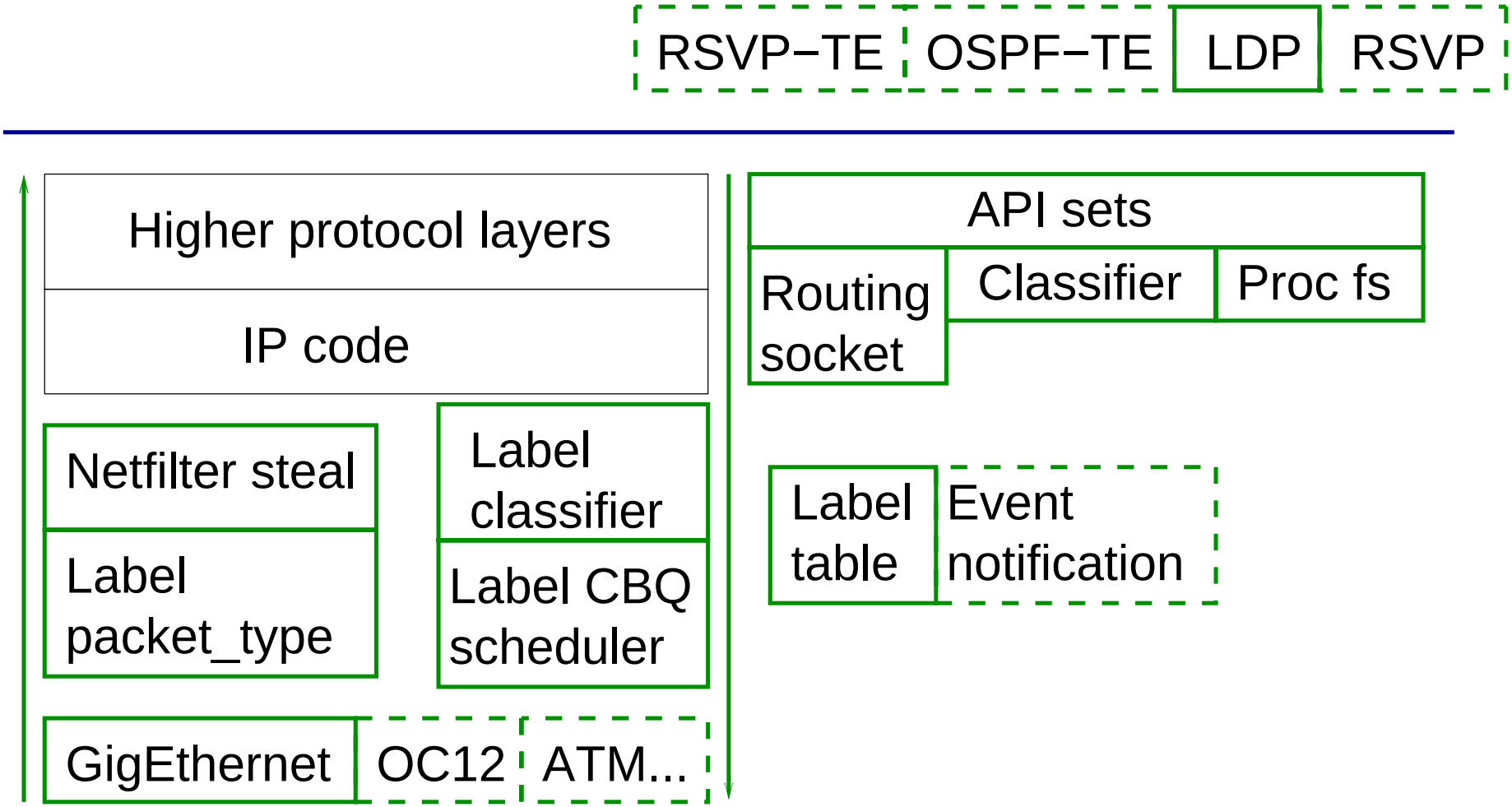
Why a layer 2.5 toolkit?

- An independent niche in the protocol stack
  - < 3, not end−to−end: no universality required
  - >2, not link−layer−specific: not "hostage" to a particular technology
  - Scope (of signaling, etc.) is as broad or narrow as you wish (at least within your area of control)
- Upper, lower layers are yours to command
  - Internally: queueuing, shaping, merging, splitting
  - Externally: routing, signaling

Why a new NIST Switch?

- Linux (2.4)−based
  - Takes advantage of all the Linux hands nowadays
- Highly modular (toolkit design)
  - No kernel rebuild (~), loadable features (queueing...)
- Improved functionality
  - Much greater label selectivity (port, protocol, ToS/Exp...)
  - Better statistics tracking
- Simpler interfaces
  - Proc FS, iptables, ip/tc

Obligatory Architecture Box Diagram



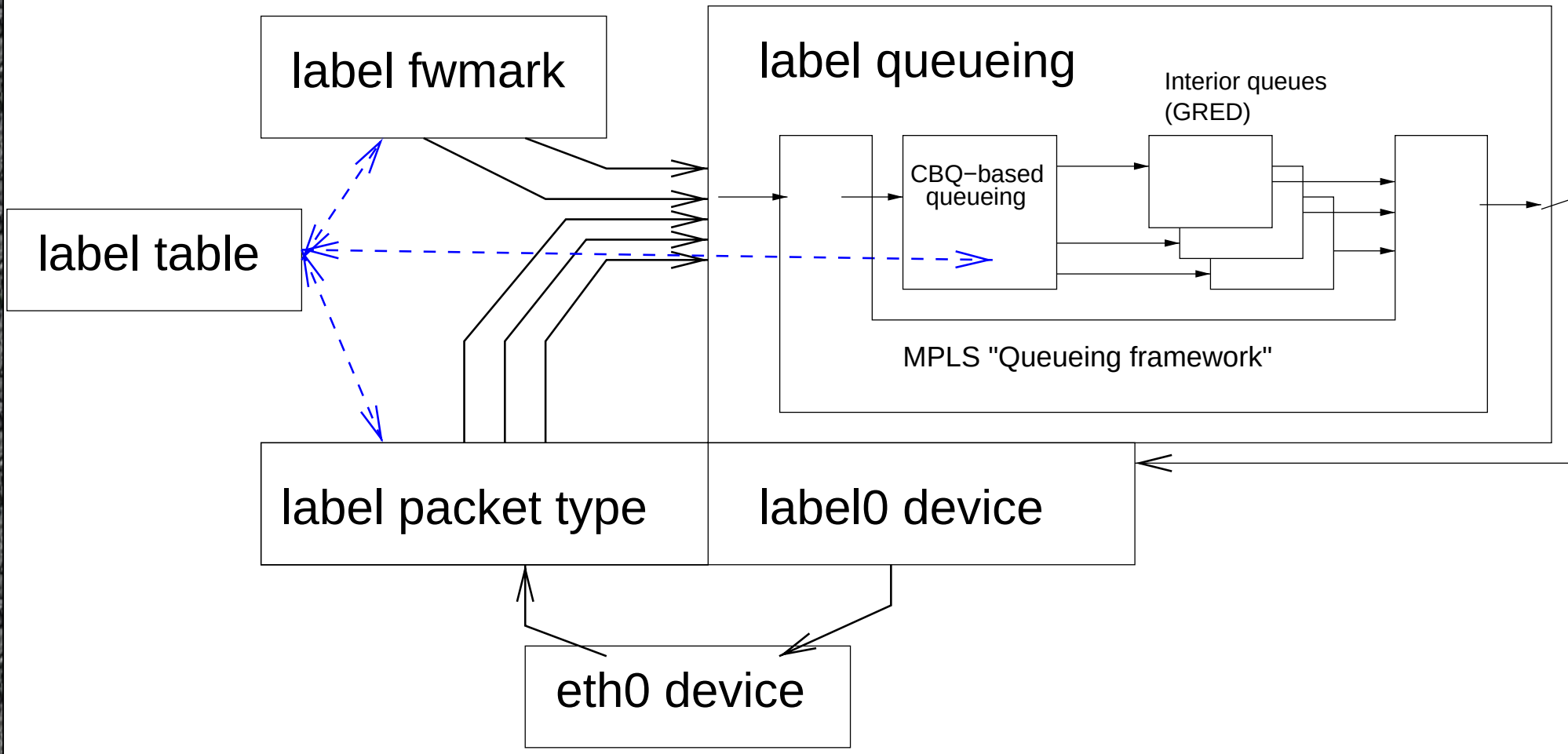
Some interesting user−level features

- Composable queueing framework
  - Currently CBQ, TBF, GRED
  - Many other potential options
  - Goals: support full−blown TE, DiffServ at edges; down to simple merge in interior
- Proc FS label access
  - "Direct" access to kernel data structures

Composable queueing framework

- Label interfaces appear as usual Linux network interfaces, layered on top of "real" interfaces
  - /dev/label*i*
  - Keep statistics, etc., by usual means
- Queueing disciplines can be pushed on top of these by ordinary Linux tools
  - ip, tc

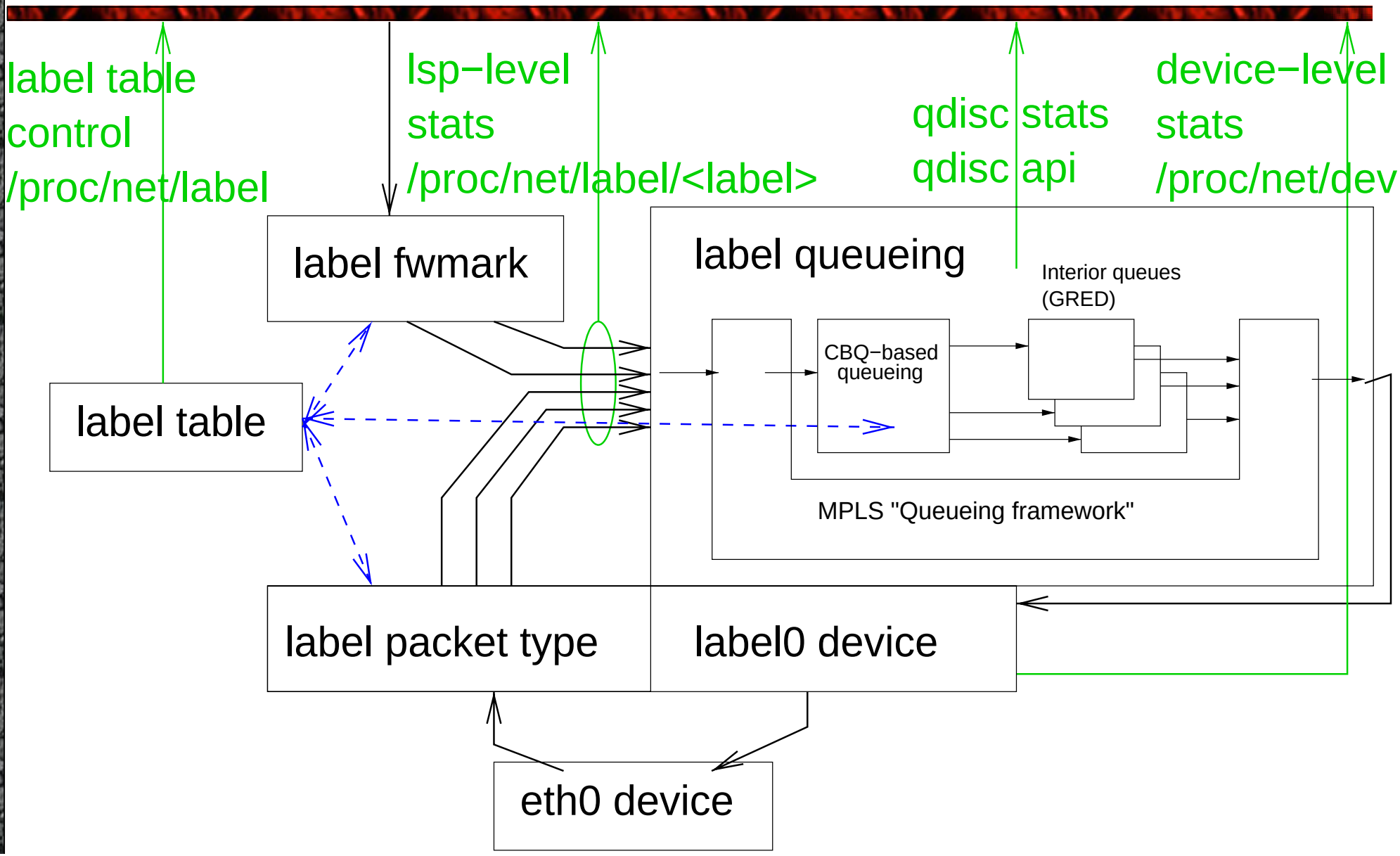
Labeled data flow and queueing



Proc FS label access

- Label table appears as directory in /proc filesystem
  - /proc/net/label
  - Label table entries are "files" within this directory
    - Name: *labelid* or *labelid.exp* (e.g., 23, 47.3)
    - Contents: formatted dump of label table data structure
- File−like appearance makes kernel data structures directly editable by ordinary tools
  - You can even vi them!

Proc FS statistics and control points



Near−term TBD list

- Validation/performance efforts
  - Calibrate queueing (on regular Linux first)
  - Profiling, tweaking
- Routing, signaling, restoration support
  - RSVP (from FreeBSD+ISI Linux) (UMBC)
  - Help move UMBC over to new platform
  - OSPF−OMP (UKansas?)
  - Event notification

Future research outlook: What can NIST Switch enable?

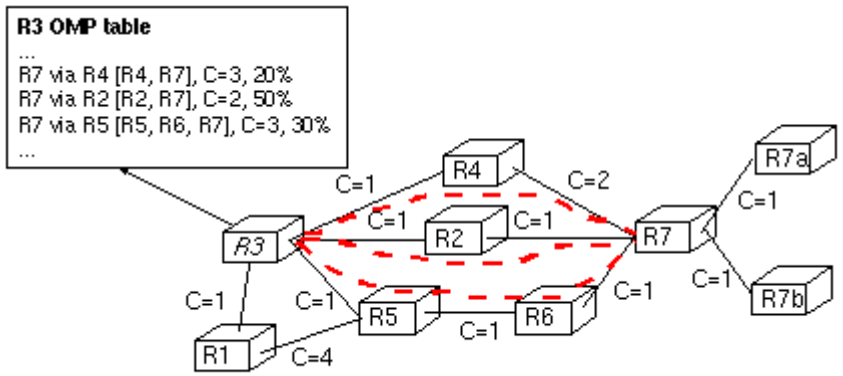
- Mid−term testbed proposal: Optimized multipath in fairly flat networks
  - Novel high−bandwidth, low latency routing
  - Uses label−based OMP as hook for investigating TE, restoration issues in simple form
  - Minimal hardware (fast Ethernet), software requirements, but has future applicability as well
  - Combines several recent trends, builds off UMBC work
  - Good example of the "layer 2.5" viewpoint

Optimized multipath in fairly flat networks

- Labeled OMP as TE, restoration tool
  - (Loosely) bind together multiple links into layer 2.5 channels
  - OMP handles bandwidth sharing, failure recovery
  - Sort of a RAID for networks
- Fairly flat networks
  - Network is multigraph with fairly small diameter (~3)
  - Testbed version: Ethernet switches with high port count (~24)

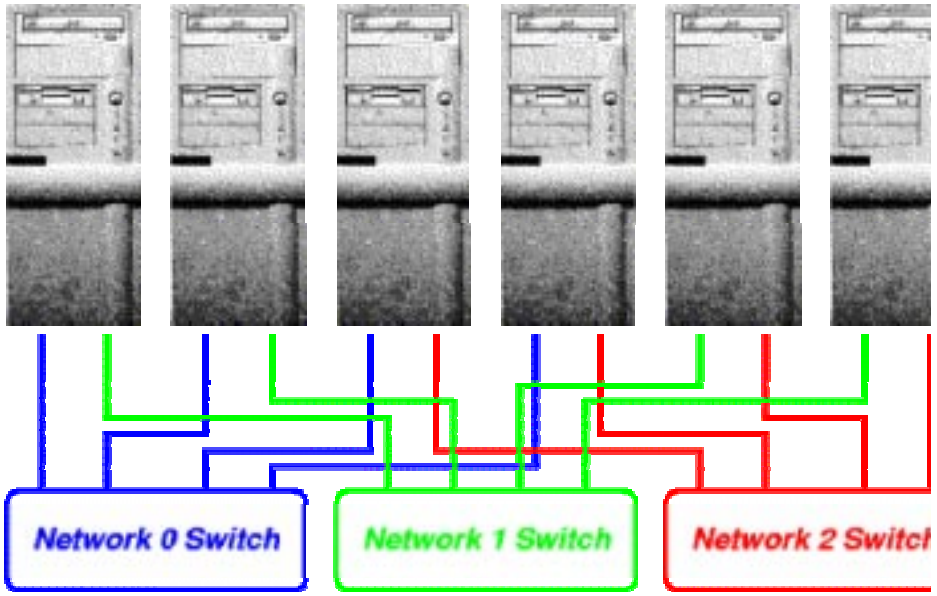
Optimized multipath

- Optimized multipath routing
  - Load−balancing scheme which allows scalable path bandwidth provisioning with (~) no fixed limits
  - Sort of a higher−level version of channel bonding
  - Sample implementation on Zebra OSPF may be available (*B. Ramachandran*, ITTC, U Kansas)
  - Labeled version needed



Fairly flat networks

- Flat neighborhood networks *(Dietz, Mattox,* University of Kentucky) for Beowulf
  - GA for designing maximal bandwidth one−hop networks
  - "Fairly flat" variant allows 2 or 3 hops
  - Requires new design algorithm (heuristic)



Label−based Ethernet switching

- Exploit capabilities of current Ethernet switches (UMBC)
  - Labels used to signal between layer 2.5 "Steiner points"
  - Intermediate layer 2 nodes controlled by manipulating forwarding tables
  - Used here to build up multipaths